

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-234263

(43)Date of publication of application : 27.08.1999

(51)Int.Cl. H04L 9/32  
G09C 1/00  
H04L 9/10

(21)Application number : 10-030053

(71)Applicant : FUJI XEROX CO LTD

(22)Date of filing : 12.02.1998

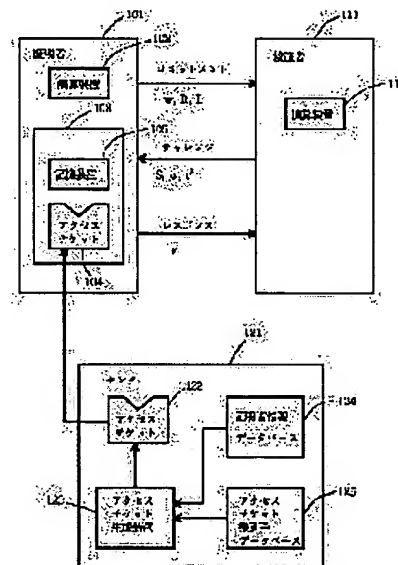
(72)Inventor : SUZUKI KOJI  
NAKAGAKI JUHEI  
SHIN YOSHIHIRO

## (54) METHOD AND DEVICE FOR MUTUAL AUTHENTICATION

## (57)Abstract:

PROBLEM TO BE SOLVED: To provide a mutual authentication suitable for a device of low arithmetic capacity.

SOLUTION: A testifier 101 generates a commitment (w) and sends it to a verifier 111. The verifier 111 sends challenges S,  $\mu$  and  $r'$  to the testifier 101 and calculates an index  $g=h(r', \mu)$  for verification while using a unidirectional hash function (h) shared with the testifier 101. The testifier 101 verifies S while using disclosed information, further generates an index  $g'=h(r', \mu)$  for verification while using  $r', \mu$  and (h), generates response generation information  $y \equiv r'F(d, L, It)g' \pmod n$  while using a unidirectional hash function F shared with a center 121, further generates a response  $y \equiv y'tg' \pmod n$  while using the index g' for verification and sends it to the verifier 111. The verifier 111 generates verification information  $w' \equiv JA gE y v \pmod n$  while using disclosed testifier information JA or the like of the testifier 101 and compares it with the commitment (w).



## LEGAL STATUS

[Date of request for examination]

20.09.2002

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the  
examiner's decision of rejection or application converted  
registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of  
rejection]

[Date of extinction of right]

## 【特許請求の範囲】

【請求項1】 証明者が有資格者であることを保証するセンターが生成する有資格情報を用い、証明者側の有資格性、および検証者側の認証の双方をゼロ知識証明方式を用いて行う相互認証方法であって、証明者が不正な操作を施すことが困難な演算方法によって生成されたレスポンス生成情報に対して、センターが発行する有資格者情報を用いて該レスポンス生成情報に演算を施し、演算を施した情報をレスポンスとして検証者に提示することにより有資格者であることを証明することを特徴とする相互認証方法。

【請求項2】 証明者側の有資格性の証明を行うためのゼロ知識証明方式として、零化域を求めることが計算量的に困難な有限アーベル群を用いたゼロ知識証明方式を用いる請求項1記載の相互認証方法。

【請求項3】 検証者側の認証を行うためのゼロ知識証明方式として、零化域を求めることが計算量的に困難な有限アーベル群を用いたゼロ知識証明方式を用いる請求項1または2記載の相互認証方法。

【請求項4】 検証者側の認証を行うためのゼロ知識証明方式において、検証者の署名を用いた認証方法を用いる請求項1、2または3記載の相互認証方法。

【請求項5】 認証者側の認証を行うためのゼロ知識証明方式として、離散対数問題が計算量的に困難な有限群を用いたゼロ知識証明方式を用いる請求項1、2、3または4記載の相互認証方法。

【請求項6】 検証者側の認証を行うためのゼロ知識証明方式として、離散対数問題が計算量的に困難な有限群を用いたゼロ知識証明方式を用いる請求項1、2、3、4または5記載の相互認証方法。

【請求項7】 検証者から送られてくるチャレンジと共に認証者に対する指示情報を送信する請求項1、2、3、4、5または6記載の相互認証方法。

【請求項8】 証明器、検証器および有資格情報生成器を有し、証明者が有資格者であることを保証する有資格情報を用い、証明者側の有資格性、および検証者側の認証の双方をゼロ知識証明方式を用いて行う相互認証装置において、

上記有資格情報生成器は、上記証明者の有資格情報を生成する手段を有し、

上記証明器は、不正な操作を施すことが困難な演算方法によってレスポンス生成情報を生成する手段と、

上記レスポンス生成情報に対して、上記有資格情報生成器が発行した有資格者情報を用いて所定の演算を施しレスポンスを生成する手段とを有し、

上記検証器は、上記レスポンスを用いて検証を行なう検証手段を有することを特徴とする相互認証装置。

【請求項9】 上記証明器は証明者の演算装置、および該演算装置に接続された、外部から観察困難な装置内に保持された、演算装置及び記憶装置からなり、上記レス

ポンス生成情報の生成に不可欠な証明者に付帯した秘密情報を、該観察困難な装置内の記憶装置に保持し、該秘密情報を用いたレスポンス生成情報の生成を該外部から観察困難な装置内の演算装置において生成する請求項8記載の相互認証装置。

【請求項10】 該外部から観察困難な装置内の演算装置をスマートカードでもって実現する請求項9記載の相互認証装置。

【請求項11】 レスポンス生成情報からレスポンスを生成する演算を行うソフトウェアに対して難読化処理を施すことにより、不正な操作を行うことが困難にする請求項8、9または10記載の相互認証装置。

【請求項12】 検証者側の演算を行うソフトウェアに対して難読化処理を施すことにより、不正な操作を行うことが困難にする請求項8、9、10または11記載の相互認証装置。

【請求項13】 外部から観測困難な装置内の情報を、検証者からの指示情報に基づいて書き換える請求項8、9、10、11または12記載の相互認証装置。

【請求項14】 上記外部から観測困難な装置内の情報として課金情報を含む請求項13記載の相互認証装置。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、個人認証の分野に属し、特に検証者の認証も同時に行う相互認証の分野に属する。

## 【0002】

【従来の技術】従来、認証を行う際にはRSA (Rivest-Shamir-Adleman) 暗号を基本とした認証方式を用いるのが通常であった。RSA暗号は、素因数分解の困難性にその安全性の根拠を置いた暗号方式で、以下の式を満たすように構成された公開鍵 $(e, n)$ 、及び秘密鍵 $d$ を用いる。

## 【0003】

【数1】  $ed \equiv 1 \pmod{\phi(n)}$

ただし、上で $n$ は二つ以上の素因数分解が困難なくらい大きな素数の積からなる合成数法数、 $\phi(n)$ は、乗法群 $(Z = nZ)^*$ の位数である。

【0004】RSA暗号を用いて認証を行う場合は、上記の秘密鍵で検証側から送信されてくるチャレンジ $c$ を暗号化して、レスポンス $r$ を生成する。つまり、証明側は

## 【0005】

【数2】  $r \equiv c^d \pmod{n}$ を計算して、検証側に送り返す。次てきたレスポンス $r$ に対して以下の計算を行い、 $r'$ を求める。

## 【0006】

【数3】  $r' \equiv r^e \pmod{n}$  証明側が公開鍵 $(e, n)$ に対応

いるならば、チャレンジ $c$ と $r'$ は一致する。なぜならば、

【0007】

【数4】 $ed \equiv 1 \pmod{\phi(n)}$

より、

【0008】

【数5】 $ed = Q\phi(n) + 1$

が成り立つ。ただし、上記で $Q$ は適当な整数である。このときオイラーの定理より、

【0009】

【数6】

$$\begin{aligned} r' &\equiv r^e \pmod{n} \\ &\equiv c^{de} \pmod{n} \\ &\equiv c^{Q\phi(n)+1} \pmod{n} \\ &\equiv c (c^{\phi(n)})^Q \pmod{n} \\ &\equiv c \pmod{n} \end{aligned}$$

となるからである。チャレンジ $c$ と $r'$ が一致するとき、認証は成功したと見なされる。

【0010】RSA暗号を用いた相互認証を行う場合には、証明側の公開鍵 $(e, n)$ および秘密鍵 $d$ 、そして検証側の認証を行うための公開鍵 $(e', n')$ および秘密鍵 $d'$ を用意する。 $(e', n')$ および $d'$ の生成条件は上記に述べた $(e, n)$ および $d$ の生成条件と同じである。そしてこれら二つの鍵ペアを用いて、証明側、検証側の双方の認証を行う。

【0011】

【発明が解決しようとする課題】RSA暗号では、秘密鍵 $d$ を法数 $n$ 程度の大きな数としなければならない。これは、秘密鍵 $d$ を小さい数とすると総当たり攻撃により、 $d$ の値を知られてしまうためである。

【0012】ところで、べき乗剰余計算 $cd \pmod{n}$ の計算量は一般に指数 $d$ の値に比例する。通常、安全性のために法数 $n$ は1024ビット程度の非常に大きな数とされるため、RSA暗号を用いた認証では、証明を行う側の計算量は膨大なものになってしまう。

【0013】個人認証などにおいて、証明側はスマートカード(ICカード)などの演算能力の低いデバイスを用いることが多い。Bruce Schneiner, Applied Cryptography (Second Edition), Wiley, 1996によれば、ワークステーション(SPARC2、米国サン・マイクロシステムズ社の商標)で、法数1024ビット・公開鍵8ビットのRSA暗号系を使って1024ビットのデータを処理する時間は、署名が0.97秒、検証が0.08秒かかっている。このため、スマートカードのような、演算能力の低いデバイスでは、認証の遅延を引き起こすという問題があった。

【0014】

【課題を解決するための手段】本発明では、相互証明に用いられる認証方式としてゼロ知識証明方式を用いる。

すなわち、本発明によれば、上述の課題を解決するために、証明者が有資格者であることを保証するセンターが生成する有資格情報を用い、証明者側の有資格性、および検証者側の認証の双方をゼロ知識証明方式を用いて行う相互認証方法において、証明者が不正な操作を施すことが困難な演算方法によって生成されたレスポンス生成情報に対して、センターが発行する有資格者情報を用いて該レスポンス生成情報に演算を施し、演算を施した情報をレスポンスとして検証者に提示することにより有資格者であることを証明するようにしている。

【0015】このゼロ知識証明方式を用いた相互認証方式では、べき乗剰余計算に用いられる指数が数ビットから数十ビットで済むため、認証に必要な計算量を大幅に削減することができ、認証の遅延を防止することが可能となる。

【0016】なお、本発明は、装置やコンピュータプログラム製品としても実現可能である。

【0017】

【発明の実施の態様】以下では、認証者が検証者に送るコミットメント $w$ として、

【0018】

【数7】 $w \equiv r^v \pmod{n}$ を送り、検証者の認証をRSA暗号を説明する。ただし、上で $n$ は素因数分解が困難な二つの大きな素数 $p, q$ の積からなる合成数法数、 $v$ は $\lambda(p-1, q-1)$ と互いに素な $v \geq 3$ を満たす数ビットから数十ビットの小さな素数とする( $\lambda(x, y)$ は $x, y$ の最小公倍数を表わす)。

【0019】以下では、センタによって認証者に発行される有資格情報をアクセスチケットと呼ぶことにする。

【0020】図1に本実施例の基本構成を示す。

【0021】図1で、証明者101は認証プロトコルを実行する演算装置102、及び証明者の身許を保証するトークン103から成る。トークン103はスマートカードなどのタンパープルーフな(tamper-proof:内部のデータや処理を外部から観察困難な状態)装置で実現され、不正な方法では書き換えができない記憶装置105、及びレスポンス生成情報を生成する際に用いられ、検証者111との間で共有される一方向性ハッシュ関数 $h$ 、同じくレスポンス生成情報を生成する際に用いられ、センタ121との間で共有される秘密の一方向性ハッシュ関数 $F$ を備えている。記憶装置105には証明者が受けたサービスに対応する課金情報などが記憶される。またトークン103には証明者101の有資格性を保証するためのアクセスチケット104が演算装置102を通して送信され、これを内部に保持することができる。

【0022】検証者111は認証プロトコルを実行する演算装置112から成る。演算装置112は、チャレンジを生成する際に用いられ、認証者101と共有される

一方向性ハッシュ関数 $h$ を保持している。

【0023】センタ121は、アクセスチケット生成装置123、証明者情報データベース124、アクセスチケット発行データベース125から成る。センタ121は証明者の要求に応じて、証明者情報データベース124、アクセスチケット発行データベース125から必要な情報を取り出し、アクセスチケット生成装置123においてアクセスチケット122を生成する。アクセスチケット生成装置123は、アクセスチケットを生成する際に用いられ、証明者101のトークン103との間で共有される秘密の一方向性ハッシュ関数 $F$ を備えている。

【0024】以下に証明者101、検証者111、センタ121が行う事前準備について説明する。

【0025】センタ121は、証明者101の電話番号、FAX番号、住所などから成る公開情報 $I_A$ を生成する。次にセンタ121は $I_A$ にISO9796などで定められる冗長性を加えて公開証明者情報 $J_A$ を生成し、これを公開する。

【0026】またセンタ121は、この $J_A$ から証明者秘密情報 $C_A$ を以下の式に従って生成する。

【0027】

【数8】 $C_A J_A^{-s} \bmod n$  ここで、 $s$ はセンタ121の次に認証の手順について説明する。認証の手順は以下のステップに従って実行される。

【0028】

【数9】 $sv \equiv 1 \bmod \lambda(p-1, q-1)$   
公開指数 $v$ は $\lambda(p-1, q-1)$ と互いに素な素数なので、上記を満たす数 $s$ が存在する。この値は、例えばユークリッドの互除法などで法 $\lambda(p-1, q-1)$ の下での $v$ の逆数を求め、これを $s$ とすればよい。

【0029】センタ121は、上式によって生成された $C_A$ を証明者101に付帯させた状態で、証明者情報データベース124に格納する。

【0030】次に検証者111は、RSA暗号に基づいた認証を行うために、検証者公開鍵 $(e_B, n')$ 、及び検証者秘密鍵 $d_B$ を生成する。ここで、 $n'$ はRSA暗号に用いる法数で、素因数分解が困難な二つの大きな素数 $p'$ 、 $q'$ の積からなる合成数法数とし、 $e_B$ は $\lambda(p'-1, q'-1)$ と互いに素な小さな数とする。 $e_B$ を小さな数とする理由は検証者の認証を行う証明者101の計算量を少なくするためである。また検証者111の秘密鍵 $d_B$ を以下の式を満たすものとする。

【0031】

【数10】

$e_B d_B \equiv 1 \bmod \lambda(p'-1, q'-1)$   
上記の式を満たす $d_B$ を求めるには、例えばユークリッドの互除法などで法 $\lambda(p'-1, q'-1)$ の下での $e_B$ の逆数を求め、これを $d_B$ とすればよい。こうして生成した公開鍵 $(e_B, n')$ を公開し、 $d_B$ を検証者11

1の秘密情報として保持する。

【0032】以上の検証者用の公開鍵、及び秘密鍵の生成はセンタがこれを行い、秘密裏に秘密鍵を検証者111へ送付してもよい。

【0033】以下に証明者101の要求に応じて、センタ121がアクセスチケット104を生成する過程を説明する。

【0034】まず、認証者101は検証者111から受けたサービスセンタ121に通知する。センタ121は証明者に対して、その証明者に付帯されたトークン秘密情報 $d$ 、及び証明者秘密情報 $C_A$ を証明者情報データベースから検索し、検証者111から受けるサービスからアクセスチケット識別子 $I_t$ 、及び証明者101に割り当てられたサービス利用制限情報 $L$ をアクセスチケット発行データベースから取り出す。サービス利用制限情報 $L$ には、例えば年齢、資格などに応じた証明者の受けることのできるサービスの制限を記載する。

【0035】センタ121は $d$ 、 $C_A$ 、 $L$ 、 $I_t$ から以下の式に従ってアクセスチケット $t$ を計算し、証明者101にインターネットなどを使って送付する。

【0036】

【数11】 $t = C_A F(d, L, I_t)^{-1}$

次に認証の手順について説明する。認証の手順は以下のステップに従って実行される。

【0037】[ステップ200] 証明者101は、トークン103において乱数 $r$ を生成する。

【0038】[ステップ201] 続いて証明者101は、ステップ200で生成した乱数 $r$ を用いてコミットメント $w$ を以下の式に従って生成する。

【0039】

【数12】 $w \equiv r^v \bmod n$  【0040】[ステップ202] は、ステップ201で生成したコミットメント $w$ を演算装置102に送信する。

【0041】[ステップ203] 続いて証明者101は、演算装置102を用いて、ステップ201で生成したコミットメント $w$ 、検証者111から受けたサービスに対応するアクセスチケット識別子 $I_t$ 、及び対応するサービス利用制限情報 $L$ を検証者111の演算装置112へ送信する。

【0042】[ステップ204] 検証者111は演算装置112において乱数 $r'$ を生成する。

【0043】[ステップ205] 続いて検証者111は演算装置112においてステップ204で生成された乱数 $r'$ 、サービスに対応する課金情報等のメッセージ $\mu$ 、アクセスチケット識別子 $I_t$ を用いて、

【0044】

【数13】 $\delta = \mu \parallel I_t \parallel r'$

を生成する。上記で「 $\parallel$ 」はビット結合を表わす。

【0045】[ステップ206] 続いて検証者111は

演算装置112において、証明者101から送られてきたコミットメントwの下位ビットを $\delta$ で置き換えることによって、 $w[\delta]$ を生成する。

【0046】[ステップ207] 続いて検証者111は演算装置112においてステップ206で生成された $w[\delta]$ を用いて、検証者111の認証データSを以下の式に従って生成する。

【0047】

$$\text{【数14】 } S \equiv w[\delta] \text{ dB mod } n'$$

【0048】[ステップ208] 続いて検証者111は演算装置112を用いて、ステップ207で生成された検証者111の認証データSと、課金情報などの証明者に対するメッセージ $\mu$ 、ステップ204で生成された乱数 $r'$ をチャレンジとして証明者101に送信する。

【0049】[ステップ209] 続いて検証者111は演算装置112を用いて、課金情報などの証明者に対するメッセージ $\mu$ 、ステップ204で生成された乱数 $r'$ 、証明者101との間で共有している一方向性ハッシュ関数hを用いて、検証用指数gを以下の式に従って生成する。

【0050】

$$\text{【数15】 } g = h(r', \mu)$$

【0051】[ステップ210] 認証者101は演算装置102を用いて、検証者111から送られてきた認証データSと、課金情報などの証明者に対するメッセージ $\mu$ 、チャレンジ $r'$ をトークン103へ送信する。

【0052】[ステップ211] 証明者101は、トークン103において、検証者111から送られてきた認証データS、及び検証者111の公開鍵 $(e_B, n')$ を用いて、以下の値 $S'$ を計算する。

【0053】

$$\text{【数16】 } S' \equiv S e_B \text{ mod } n'$$

【0054】[ステップ212] 続いて証明者101は、トークン103において、ステップ211で生成した $S'$ の上位ビットとステップ201で生成したwの上位ビットを比較し、同じ場合は検証者の認証が成功したものとしてステップ213へ進む。異なる場合は、不正な検証者と見なして認証の手順を中止する。

【0055】[ステップ213] 続いて証明者101は、トークン103において、検証者111から送られてきた乱数 $r'$ 、課金情報などの証明者に対するメッセージ $\mu$ 、及び証明者101と共有している一方向性ハッシュ関数hを用いて、検証用指数 $g'$ を以下の式に従って生成する。

【0056】

$$\text{【数17】 } g' = h(r', \mu)$$

【0057】[ステップ214] 続いて証明者101は、トークン103において、ステップ200で生成した乱数r、ステップ213で生成した検証用指数 $g'$ 、  

$$w' \equiv J_A g y^v \text{ mod } n$$

トークン103の記憶装置105内に記憶された証明者の秘密情報d、サービス利用制限情報L、アクセスチケット識別子 $I_t$ 、センタ121と共有している秘密の一方向性ハッシュ関数Fを用いて、以下の式に従ってレスポンス生成情報 $y'$ を生成する。

【0058】

$$\text{【数18】 } y' \equiv r F(d, L, I_t) g' \text{ mod } n \text{ 【0059】}$$

は、ステップ213で生成した検証用指数 $g'$ 、及びステップ214で生成した $y'$ をトークン103から演算装置102へ送信する。

【0060】[ステップ216] 続いて証明者101は、検証者111から送られてきた課金情報などの証明者に対するメッセージ $\mu$ に基づき、トークン103内の記憶装置105内の情報を書き換える。

【0061】[ステップ217] 続いて証明者101は、演算装置102において、ステップ214で生成した $y'$ 、アクセスチケットt、及びステップ213で生成した検証用指数 $g'$ を用いて、以下の式に従ってレスポンスyを生成する。

【0062】

$$\text{【数19】 } y \equiv y' t g' \text{ mod } n \text{ 【0063】 [ステップ21}$$

は、ステップ217で生成したyを演算装置102から検証者111の演算装置112へ送信する。

【0064】[ステップ219] 検証者111は、演算装置112において、証明者101の公開証明者情報 $J_A$ 、ステップ209で生成した検証用指数g、証明者101から送られてきたレスポンスy、及び公開指数vを用いて、以下の式に従って検証情報 $w'$ を生成する。

【0065】

$$\text{【数20】 } w' \equiv J_A g y^v \text{ mod } n \text{ 【0066】 [ステップ}$$

は、演算装置112において、証明者101から送られてきたコミットメントwとステップ219で生成した検証情報 $w'$ を比較し、二つが一致すれば証明者101の認証に成功したものとし、一致しなければ証明者101の認証に失敗したものとする。

【0067】以下に、正当な証明者が正当なアクセスチケットを用いた場合には、検証者が計算する $w'$ と証明者が生成したコミットメントwとが一致することを示す。

【0068】証明者及び検証者上記のステップに従って計算をすると、明らかに

【0069】

$$\text{【数21】 } g = g'$$

が成立するので、

【0070】

【数22】

$$\begin{aligned} &\equiv J_A \cdot g^{r^v F(d, L, I_t)} \cdot g^{v C_A} \cdot g^{v F(d, L, I_t)} \cdot g^{-v} \\ \text{mod } n &\equiv J_A \cdot g^{r^v C_A} \cdot g^v \text{ mod } n \equiv r^v (J_A C_A^v) \cdot g \text{ mod } n \end{aligned}$$

となる。ここで、s の条件より、

【0071】

【数23】  $s \cdot v \equiv 1 \text{ mod } \lambda(p-1, q-1)$

が成り立っている。これにより、

【0072】

【数24】  $s \cdot v = Q \cdot \lambda(p-1, q-1) + 1$

が成立する。ここで、上式でQは適当な整数である。よって、

【0073】

【数25】

$$\begin{aligned} J_A^{-sv} &= (J_A^{-1})^{Q \cdot \lambda(p-1, q-1) + 1} \\ &\equiv J_A^{-1} (J_A^{-Q})^{\lambda(p-1, q-1)} \text{ mod } n \end{aligned}$$

ここで、中国人剰余定理 (Chinese Remainder Theorem)、及びオイラーの定理より、

【0074】

【数26】

$$(J_A^{-Q})^{\lambda(p-1, q-1)} \equiv 1 \text{ mod } n$$

が成り立つので

【0075】

【数27】  $J_A^{-sv} \equiv J_A^{-1} \text{ mod } n$  が成立する。よって

【0076】

【数28】

$$w' \equiv r^v (J_A J_A^{-sv}) \cdot g \text{ mod } n \equiv r^v (J_A J_A^{-1}) \cdot g \text{ mod } n$$

【0077】上記の方法では、ゼロ知識証明方式を用いることにより証明者側の計算は数ビットから数十ビットの小さなべき指数に関するべき乗剰余計算を主体とした計算しか行わないため、大きなべき指数に関するべき乗剰余計算を行うRSA暗号を用いた方式に比較して、高速な認証が可能になることがわかる。

【0078】上記では、証明者側の認証方式においてv

をλと互いに素な  $v \geq 3$  を満たす素数としたが、これを  $v = 2$  としても同じ効果が得られる。

【0079】また、証明者側の認証方式、及び検証者側の認証方式として合成数nを法とする有理整数環の剰余類環の乗法群  $(Z = nZ)^*$  上で定義された演算を用いたゼロ知識証明、あるいはSchorr署名方式を用いたゼロ知識証明方式、あるいは楕円曲線上で定義された有限群を用いたゼロ知識証明方式、あるいは素数pを法とする有理整数環の剰余類環の乗法群  $(Z = pZ)^*$  を用いたゼロ知識証明方式を用いても同様な効果が得られる。

【0080】

【発明の効果】以上説明したように、ゼロ知識証明方式を用いた相互認証方式では、べき乗剰余計算に用いられる指数が数ビットから数十ビットで済むため、認証に必要な計算量を大幅に削減することができ、認証の遅延を防止することが可能となる。

【図面の簡単な説明】

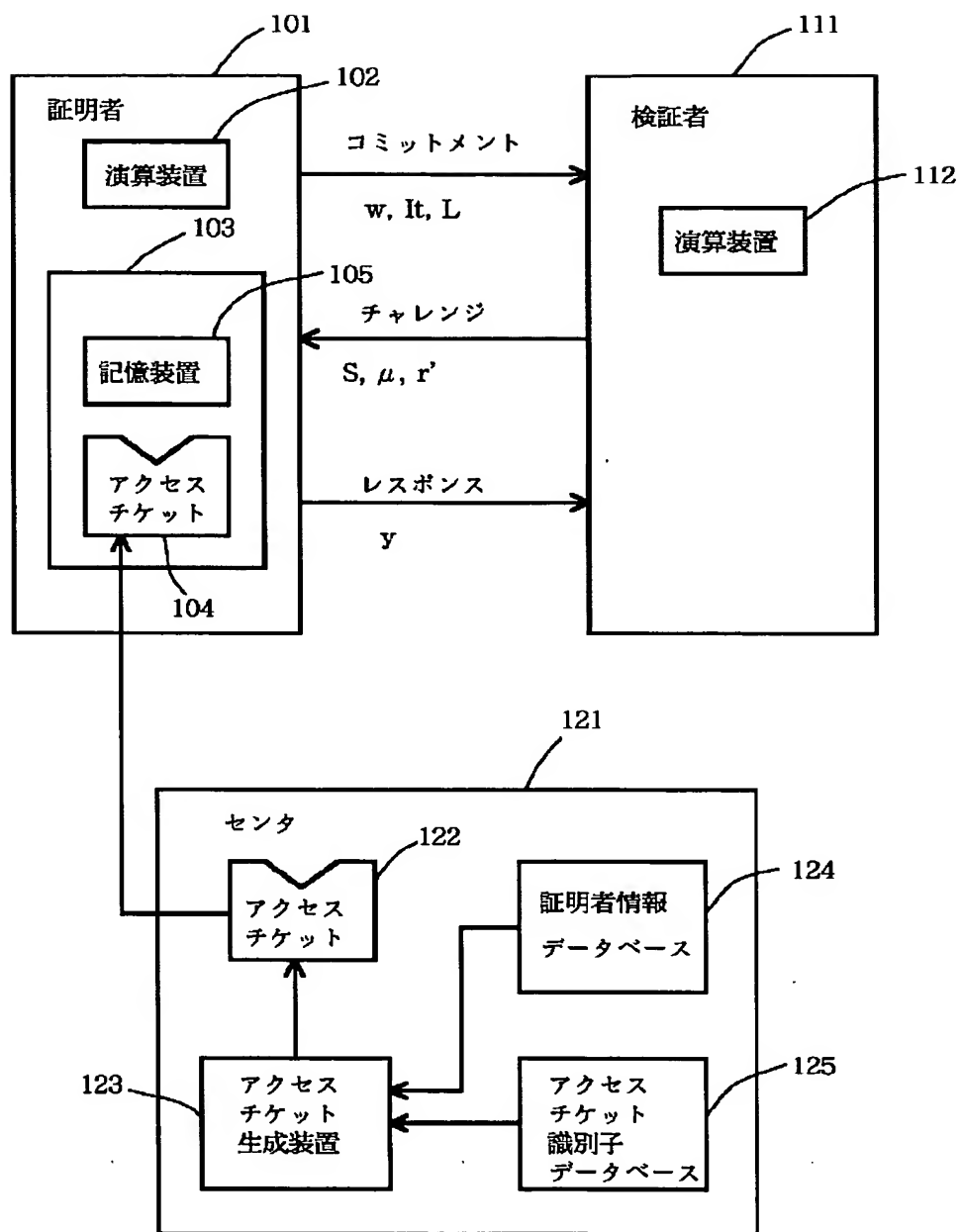
【図1】 本発明の実施例の構成を示すブロック図である。

【図2】 上述実施例の動作を説明するフローチャートである。

【符号の説明】

101	証明者
102	演算装置
103	トークン
104	アクセスチケット
105	記憶装置
111	検証者
112	演算装置
121	センタ
122	アクセスチケット
123	アクセスチケット生成装置
124	証明者情報データベース
125	アクセスチケット発行データベース

【図1】



【図2】

